**CSCI 360 Shift Instructions**

The result of performing an *n* position right logical shift on a binary number containing *m* digits is obtained by:

1. Removing the RIGHTMOST *n* digits from the original number
2. Shifting the remaining digits *n* positions to the right
3. Placing *n* zeros to the left of the resulting number

For example, performing a three position, right logical shift on

the number 10110001 results in:

BEFORE: 1 0 1 1 0 0 0 1

AFTER: 0 0 0 1 0 1 1 0

A left shift is performed in a similar manner.

Performing a three position, left logical shift on the number

10110001 results in:

BEFORE: 1 0 1 1 0 0 0 1

AFTER: 1 0 0 0 1 0 0 0

**Shift Left Logical**

Format: label SLL R,D(B)

The content of the register specified by R are shifted to the left depending on the rightmost six bits of the calculated D(B) address.

The condition code is not altered.

The digits that are shifted off are lost.

Suppose that register 7 has the value 0F 0F 0F 0F. Execution of:

SLL R7,2

0 F 0 F 0 F 0 F

BEFORE: 0000 1111 0000 1111 0000 1111 0000 1111

AFTER: 0011 1100 0011 1100 0011 1100 0011 1100

3 C 3 C 3 C 3 C

will change the contents of register 7 to 3C 3C 3C 3C.

Suppose that register 7 has the value 0F 0F 0F 0F and that

register 2 has the value 00 00 00 05. Execution of:

SLL R7,0(R2)

0(R2) = 0 + 000005 = 000005

The rightmost 6 bits of 0(R2) are 000101. When converted to

decimal this is 5.

0 F 0 F 0 F 0 F

BEFORE: 0000 1111 0000 1111 0000 1111 0000 1111

AFTER: 1110 0001 1110 0001 1110 0001 1110 0000

E 1 E 1 E 1 E 0

will change the contents of register 7 to E1 E1 E1 E0.

**Shift Right Logical**

Format: label SRL R,D(B)

The content of the register specified by R are shifted to the right depending on the rightmost six bits of the calculated D(B) address.

The condition code is not altered.

The digits that are shifted off are lost.

Suppose that register 7 has the value 0F 0F 0F 0F. Execution of:

SRL R7,6

0 F 0 F 0 F 0 F

BEFORE: 0000 1111 0000 1111 0000 1111 0000 1111

AFTER: 0000 0000 0011 1100 0011 1100 0011 1100

0 0 3 C 3 C 3 C

will change the contents of register 7 to 00 3C 3C 3C.

Suppose that register 7 has the value 0F 0F 0F 0F and that

register 2 has the value 00 00 00 05. Execution of:

SRL R7,0(R2)

0(R2) = 0 + 000005 = 000005

The rightmost 6 bits of 0(R2) are 000101. When converted to

decimal this is 5.

0 F 0 F 0 F 0 F

BEFORE: 0000 1111 0000 1111 0000 1111 0000 1111

AFTER: 0000 0000 0111 1000 0111 1000 0111 1000

0 0 7 8 7 8 7 8

will change the contents of register 7 to 00 78 78 78.

**Shift Left Double Logical**

Format: label SLDL R,D(B)

The content of the even-odd register pair specified by R are shifted to the left depending on the rightmost six bits of the calculated D(B) address.

The condition code is not altered.

The digits that are shifted off are lost.

The number being shifted is treated as a 64 bit number.

Suppose that register 6 has the value 00 00 F0 F0 and register 7

has the value 0F 0F 0F 0F. Execution of:

SLDL R6,12

BEFORE: 0000 0000 0000 0000 1111 0000 1111 0000

0000 1111 0000 1111 0000 1111 0000 1111

AFTER: 0000 1111 0000 1111 0000 0000 1111 0000

1111 0000 1111 0000 1111 0000 0000 0000

will change the contents of register 6 to 0F 0F 00 F0 and

register 7 to F0 F0 F0 00.

**Shift Right Double Logical**

Format: label SRDL R,D(B)

The content of the even-odd register pair specified by R are shifted to the right depending on the rightmost six bits of the calculated D(B) address.

The condition code is not altered.

The digits that are shifted off are lost.

The number being shifted is treated as a 64 bit number.

Suppose that register 6 has the value 00 00 F0 F0 and register 7

has the value 0F 0F 0F 0F. Execution of:

SRDL R6,6

BEFORE: 0000 0000 0000 0000 1111 0000 1111 0000

0000 1111 0000 1111 0000 1111 0000 1111

AFTER: 0000 0000 0000 0000 0000 0011 1100 0011

1100 0000 0011 1100 0011 1100 0011 1100

will change the contents of register 6 to 00 00 03 C3 and

register 7 to C0 3C 3C 3C.

The logical shift instructions are mainly used when you want to get rid of bits on the beginning or end of a number. They are also used when you want to position the bits in a register.

**Shift Left Arithmetic**

Format: label SLA R,D(B)

The content of the register specified by R are shifted to the left depending on the rightmost six bits of the calculated D(B) address.

An arithmetic left shift is equivalent to multiplying by a power of 2. The power of 2 to multiply by is specified by the second operand.

The first (or sign) bit (bit 0) does not participate in the shift.

If the bit shifted out of position 1 does not match the sign bit, overflow will occur. Therefore, a valid arithmetic shift will never alter the sign bit.

The condition code is set:

Code Meaning

0 Result is 0

1 Result is less than 0

2 Result is greater than 0

3 Overflow

Suppose that register 7 has the value 00 00 00 05. Execution of:

SLA R7,2

0 0 0 0 0 0 0 5

BEFORE: 0000 0000 0000 0000 0000 0000 0000 0101

AFTER: 0000 0000 0000 0000 0000 0000 0001 0100

0 0 0 0 0 0 1 4

will change the contents of register 7 to 00 00 00 14.

**Shift Right Arithmetic**

Format: label SRA R,D(B)

The content of the register specified by R are shifted to the right depending on the rightmost six bits of the calculated D(B) address.

An arithmetic right shift is equivalent to dividing by a power of 2. The power of 2 to divide by is specified by the second operand.

The value of bits that are filled in on the left are equal to the sign bit. If the number is positive, the leftmost bits will be filled with zero. If the number is negative, the leftmost bits will be filled with ones.

The condition code is set:

Code Meaning

0 Result is 0

1 Result is less than 0

2 Result is greater than 0

Suppose that register 7 has the value FF FF FF FF. Execution of:

SRA R7,2

F F F F F F F F

BEFORE: 1111 1111 1111 1111 1111 1111 1111 1111

AFTER: 1111 1111 1111 1111 1111 1111 1111 1111

F F F F F F F F

will change the contents of register 7 to FF FF FF FF.

Remember that this is integer division.

Suppose that register 7 has the value 00 00 00 0C.

Execution of:

SRA R7,2

0 0 0 0 0 0 0 C

BEFORE: 0000 0000 0000 0000 0000 0000 0000 1100

AFTER: 0000 0000 0000 0000 0000 0000 0000 0011

0 0 0 0 0 0 0 3

will change the contents of register 7 to 00 00 00 03.

**Shift Left Double Arithmetic**

Format: label SLDA R,D(B)

The content of the even-odd register pair specified by R are shifted to the left depending on the rightmost six bits of the calculated D(B) address.

An arithmetic left shift is equivalent to multiplying by a power of 2. The power of 2 to multiply by is specified by the second operand.

The number being shifted is treated as a 64 bit number.

The first (or sign) bit (bit 0) does not participate in the shift.

If the bit shifted out of position 1 does not match the sign bit, overflow will occur. Therefore, a valid arithmetic shift will never alter the sign bit.

The condition code is set:

Code Meaning

0 Result is 0

1 Result is less than 0

2 Result is greater than 0

3 Overflow

Suppose that register 6 has the value 00 00 00 00 and register 7

has the value 00 00 00 0A. Execution of:

SLDA R6,3

BEFORE: 0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 1010

AFTER: 0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0101 0000

will change the contents of register 6 to 00 00 00 00 and

register 7 to 00 00 00 50.

**Shift Right Double Arithmetic**

Format: label SRDA R,D(B)

The content of the even-odd register pair specified by R are shifted to the right depending on the rightmost six bits of the calculated D(B) address.

An arithmetic right shift is equivalent to dividing by a power of 2. The power of 2 to divide by is specified by the second operand.

The value of bits that are filled in on the left are equal to the sign bit. If the number is positive, the leftmost bits will be filled with zero. If the number is negative, the leftmost bits will be filled with ones.

The condition code is set:

Code Meaning

0 Result is 0

1 Result is less than 0

2 Result is greater than 0

Suppose that register 6 has the value 00 00 00 00 and register 7

has the value 00 00 01 00. Execution of:

SRDA R6,2

BEFORE: 0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0001 0000 0000

AFTER: 0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0100 0000

will change the contents of register 6 to 00 00 00 00 and

register 7 to 00 00 00 40.